

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PUB. NO.: 09-212358 [JP 9212358 A]

PUBLISHED: August 15, 1997 (19970815)

INVENTOR(s): NISHIMOTO JUNICHI

MAEJIMA HIDEO

APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP
(Japan)

APPL. NO.: 08-015016 [JP 9615016]

FILED: January 31, 1996 (19960131)

INTL CLASS: [6] G06F-009/38; G06F-009/38

JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

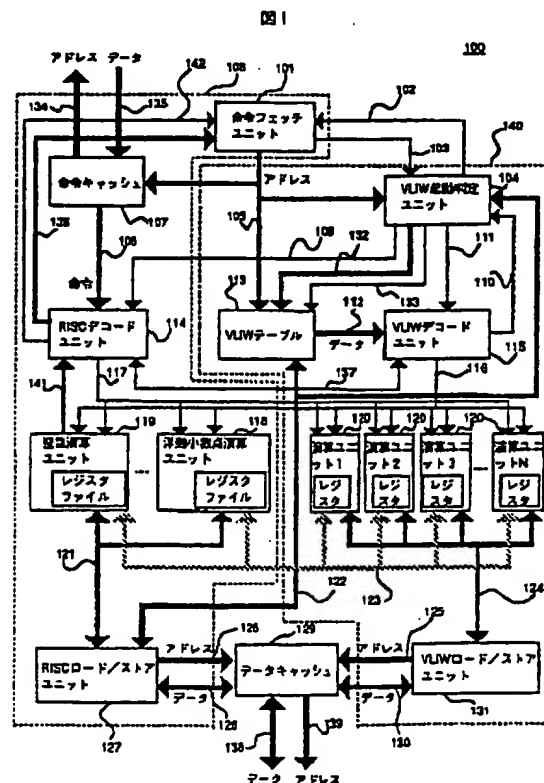
JAPIO KEYWORD: R011 (LIQUID CRYSTALS); R131 (INFORMATION
PROCESSING --

Microcomputers & Microprocessors)

ABSTRACT

PROBLEM TO BE SOLVED: To realize a processor with high performance by making a 2nd processor execute instructions of a 2nd instruction set when a 1st processor executes a specific instruction in a 1st instruction set.

SOLUTION: The microprocessor 100 uses an RISC core 106 for the execution of RISC instructions and a VLIW core 140 for the execution of VLIW instructions. An instruction fetch unit 101 outputs the address of an instruction to be executed to an instruction address bus 105. If a branch instruction (a part of RISC instruction) for the execution of VLIW instructions is generated during the execution of RISC instructions, an RISC decoding unit 114 informs the instruction fetch unit 101 of the branching generation. The VLIW instructions are read out of a VLIW table 113. A VLIW decoding unit 115 decodes the instructions to control a VLIW arithmetic unit 120 or integer arithmetic unit 119 and a floating-point unit 118 in parallel.



【特許請求の範囲】

【請求項1】第1の命令セットの命令を実行する第1のプロセッサと、第1の命令セットとは異なる第2の命令セットの命令を実行する第2のプロセッサとを有するデータ処理装置であって、

前記第1のプロセッサが前記第1の命令セットの中の所定命令を実行したとき、前記第2のプロセッサが前記第2の命令セットの命令を実行することを特徴とするデータ処理装置。

【請求項2】前記第1の命令セットの命令は可変長であり、前記第2の命令セットの命令は固定長であることを特徴とする請求項1に記載のデータ処理装置。

【請求項3】前記第1の命令セットの命令の最大語長は、前記第2の命令の語長より長いことを特徴とする請求項2に記載のデータ処理装置。

【請求項4】前記第1の命令セットの命令の最大語長は、前記第2の命令の語長より短いことを特徴とする請求項2に記載のデータ処理装置。

【請求項5】前記第1の命令セットの命令と前記第2の命令セットの命令は共に固定長であることを特徴とする請求項1に記載のデータ処理装置。

【請求項6】前記第1の命令セットの命令の語長は、前記第2の命令の語長より短いことを特徴とする請求項5に記載のデータ処理装置。

【請求項7】記憶装置に記憶されるプログラムを実行するデータ処理装置であって、第1の命令セットの命令によって構成されるプログラムと、第1の命令セットと異なる第2の命令セットの命令によって構成されるプログラムとが前記データ処理装置のアドレス空間に配置されることを特徴とするデータ処理装置。

【請求項8】前記第1の命令セットの命令は可変長であり、前記第2の命令セットの命令は固定長であることを特徴とする請求項7に記載のデータ処理装置。

【請求項9】前記第1の命令セットの命令の最大語長は、前記第2の命令の語長より長いことを特徴とする請求項8に記載のデータ処理装置。

【請求項10】前記第1の命令セットの命令の最大語長は、前記第2の命令の語長より短いことを特徴とする請求項8に記載のデータ処理装置。

【請求項11】前記第1の命令セットの命令と前記第2の命令セットの命令は共に固定長であることを特徴とする請求項7に記載のデータ処理装置。

【請求項12】前記第1の命令セットの命令の語長は、前記第2の命令の語長より短いことを特徴とする請求項11に記載のデータ処理装置。

【請求項13】演算器を有し、第1の命令群を格納する第1の記憶部から命令を読み出して実行する第1のデータ処理部と、
並列動作可能な複数の演算器を有し、同時に実行可能な命令を1命令として実行する第2のデータ処理部と、

前記第2のデータ処理部が実行する命令群を格納する第2の記憶部とを具備するデータ処理装置であって、
前記第1のデータ処理部が読み出した命令が所定の命令であるとき、前記第2のデータ処理部で実行する命令を前記第2の記憶部から読み出すことを特徴とするデータ処理装置。

【請求項14】RISC命令を実行するRISCコアと、VLIW命令を格納するVLIWテーブルと、VLIW命令を実行するVLIWコアと、前記RISCコアとVLIWコアとの動作切り替えを制御するVLIW起動判定ユニットとを有するマイクロプロセッサであって、

前記RISCコアがRISC命令のうちサブルーチンコール若しくは分岐命令を実行したとき、分岐先アドレスで前記VLIWテーブルからVLIW命令を読み出し、前記VLIW起動判定ユニットが前記RISCコアから前記VLIWコアへ動作を移し、前記VLIWテーブルからVLIW命令の完了コードが読み出されるまでVLIWコアが動作するマイクロプロセッサ。

【請求項15】請求項14のマイクロプロセッサは、さらに分岐先アドレスをアドレス変換するTLBを有し、前記VLIWテーブルから読み出されるVLIW命令を拡張するための信号をアドレス変換の際にTLBから同時によみだせるようにしたマイクロプロセッサ。

【請求項16】請求項14のマイクロプロセッサは、さらにVLIW命令へ分岐後、引き続き実行されるVLIW命令の動作とVLIW命令実行後に復帰するRISC命令の動作に関連性があるかを調べる回路を有し、関連性がない場合にVLIW命令の完了を待たずに復帰先のRISC命令を並列に実行可能なマイクロプロセッサ。

【請求項17】請求項14のマイクロプロセッサにおいて、VLIW命令の実行を禁止したり、実行を制御するレジスタをプロセッサ内部に有し、VLIW命令実行禁止時にはVLIW命令と同等の動作を行うRISC命令列に分岐することで、VLIWコアの動作を停止させ低消費電力化を図る機能を有するマイクロプロセッサ。

【請求項18】請求項14のVLIWテーブルを不揮発性メモリに内蔵するマイクロプロセッサ。

【請求項19】請求項18の不揮発性メモリは、マスクROM、フラッシュメモリ、強誘電体メモリのうちのいずれか1つであるマイクロプロセッサ。

【請求項20】請求項14のVLIWテーブルを揮発性メモリに内蔵するマイクロプロセッサ。

【請求項21】請求項20のVLIWテーブルを揮発性メモリは、SRAM或いはDRAMのうちのいずれかであるマイクロプロセッサ。

【請求項22】請求項14のVLIWテーブルをRISCコアのプロセッサで書き換え可能なメモリに内蔵するマイクロプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、マイクロプロセッサやマイクロコンピュータ等のデータ処理装置に係わり、特にソフトウェアの互換を維持して高速性能を達成するデータ処理装置に適用して有効な技術に関する。

【0002】

【従来の技術】マイクロプロセッサのアーキテクチャにはさまざまなものが存在する。例えばCISC (Complexed Instruction Set Computer) 型マイクロプロセッサ、RISC (Reduced Instruction Set Computer) 型マイクロプロセッサ、VLIW (Very Long Instruction Word) (又はLIW (Long Instruction Word) 型マイクロプロセッサなどが挙げられる。

【0003】CISC型マイクロプロセッサは、例えば8ビット或いは16ビットを最小命令長単位とし、最小命令長単位の整数倍長の可変長の命令フォーマットで命令体系(命令セット)を構成している。最大命令長は48ビット長、64ビット長、80ビット長等である。命令セットを構成する命令数が多く、1命令が複数の処理を行い、また命令語長が可変である、マイクロプログラム制御等のため、ハードウェア量が多く高速性能を追及するには難点のあるアーキテクチャであるとされている。CISC型マイクロプロセッサの例としては、モトローラ社の680X0、インテル社の80X86等がある。

【0004】高速性能を追及する一つの解として、RISC型マイクロプロセッサが出てきた。RISC型マイクロプロセッサは、16ビット長、32ビット長或いは64ビット長等の固定長の命令フォーマットで、1命令は単純な処理を行い、比較的少ない命令数で命令体系を構成している。命令が単純であるため、マイクロプロセッサの制御部のハードウェア量もCISC型のマイクロプロセッサと比較して少なく、またパイプライン処理にも適しており、高速性能が比較的出しやすいアーキテクチャである。なお、16ビット長命令と32ビット長命令が混在した命令フォーマットで命令体系を構成したものもある。RISCプロセッサの例としては、SPARC、MIPS社のR3000等がある。

【0005】最近、高速化のため、動作周波数を向上させる方法以外に同時に動作させる演算ユニットの数を増やす方法が取られてきている。この複数の演算ユニットを同時に実行する方式は、スーパースカラ方式と呼ばれている。しかし、スーパースカラ方式で、複数の命令が同時に実行可能かどうかを調べる、すなわち命令依存性を調べる機能が必要となり回路規模が増大する傾向にある。

【0006】これに対して固定長ではあるが長い命令長の命令フォーマットで命令体系を構成するプロセッサとして、VLIW型プロセッサが存在する。VLIWプロセッサでは128ビット以上の長い命令を使って複数の演算器が同時に動作する。VLIW型プロセッサではコ

ンパイル時にあらかじめ命令の依存性をチェックし同時に実行出来る命令をまとめて1命令とする。VLIW型プロセッサでは実行時に改めて命令の依存性をチェックする必要がないため、RISC型マイクロプロセッサのスーパースカラ方式に比べて回路規模は少なくなる。

なお、VLIW型プロセッサの例としては、マルチフロー・コンピュータ社のTrace ce/300等がある。

【0007】一方、RISC型のアーキテクチャの採用等により動作周波数の向上や同時に実行する命令数の増加で、マイクロプロセッサの性能が向上してきた。そのため、従来専用ハードウェアや専用コントローラを使用して行っていた、動画像の圧縮/伸長(MPEGデコーダ/エンコーダ)や三次元グラフィック処理をソフトウェアで行うことが考えられている。

【0008】MPEGデコーダ/エンコーダや三次元グラフィック処理等の同一処理の繰り返しで多量のデータを扱ういわゆるマルチメディア用途の処理にはVLIW型マイクロプロセッサが適しているとされてきている。

【0009】

【発明が解決しようとする課題】一方、従来のマイクロプロセッサ或いはCPU(中央処理装置)はCISC型又はRISC型で、それぞれのソフトウェア資産の蓄積がある。CISC型、RISC型、VLIW型のマイクロプロセッサは、それぞれ命令セット、命令フォーマット、アドレッシングモード、プログラミングモデル等は異なり、ソフトウェアの互換性はない。マイクロプロセッサは、同一アーキテクチャ(命令体系を含む。)では性能向上に限界が出てくる。そこで、アーキテクチャ変更を要求されるが、これまで築き上げてきたソフトウェア資産の継承が問題となる。

【0010】本発明の目的は、従来のソフトウェアと互換性を維持しながら、新しいアーキテクチャ(命令体系)のソフトウェアを実行する高性能なプロセッサを実現することである。

【0011】本発明の別の目的は、プロセッサ内部の資源たとえば演算ユニットなどの共有化又は共用化によるハードウェアの増加を抑えて高速性能を実現することである。

【0012】さらに、本発明の別の目的は、プロセッサ間の動作制御方式による高速化と低消費電力化の双方を実現することである。

【0013】本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【0014】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【0015】データ処理装置は、第1の命令セットの命令を実行する第1のプロセッサと、第1の命令セットと

は異なる第2の命令セットの命令を実行する第2のプロセッサとを有し、前記第1のプロセッサが前記第1の命令セットの中の所定命令を実行したとき、前記第2のプロセッサが前記第2の命令セットの命令を実行する。記憶装置に記憶されるプログラムを実行するデータ処理装置は、第1の命令セットの命令によって構成されるプログラムと、第1の命令セットと異なる第2の命令セットの命令によって構成されるプログラムとを前記データ処理装置のアドレス空間に配置する。

【0016】前記第1の命令セットの命令は可変長であり、前記第2の命令セットの命令は固定長にする。前記第1の命令セットの命令の最大語長は、前記第2の命令の語長より長い場合がある。前記第1の命令セットの命令の最大語長は、前記第2の命令の語長より短い場合がある。

【0017】前記第1の命令セットの命令と前記第2の命令セットの命令は共に固定長である場合がある。前記第1の命令セットの命令の語長は、前記第2の命令の語長より短い場合がある。

【0018】データ処理装置は、演算器を有し、第1の命令群を格納する第1の記憶部から命令を読み出して実行する第1のデータ処理部と、並列動作可能な複数の演算器を有し、同時に実行可能な命令を1命令として実行する第2のデータ処理部と、前記第2のデータ処理部が実行する命令群を格納する第2の記憶部とを具備し、前記第1のデータ処理部が読み出した命令が所定の命令であるとき、前記第2のデータ処理部で実行する命令を前記第2の記憶部から読み出す。

【0019】マイクロプロセッサは、RISC命令を実行するRISCコアと、VLIW命令を格納するVLIWテーブルと、VLIW命令を実行するVLIWコアと、前記RISCコアとVLIWコアとの動作切り替えを制御するVLIW起動判定ユニットとを有し、前記RISCコアがRISC命令のうちサブルーチンコール若しくは分岐命令を実行したとき、分岐先アドレスで前記VLIWテーブルからVLIW命令を読み出し、前記VLIW起動判定ユニットが前記RISCコアから前記VLIWコアへ動作を移し、前記VLIWテーブルからVLIW命令の完了コードが読み出されるまでVLIWコアが動作する。

【0020】前記マイクロプロセッサは、さらに分岐先アドレスをアドレス変換するTLBを有し、前記VLIWテーブルから読み出されるVLIW命令を拡張するための信号をアドレス変換の際にTLBから同時に読み出せる。

【0021】前記マイクロプロセッサは、さらにVLIW命令へ分岐後、引き続き実行されるVLIW命令の動作とVLIW命令実行後に復帰するRISC命令の動作に関連性があるかを調べる回路を有し、関連性がない場合にVLIW命令の完了を待たずに復帰先のRISC命

令を並列に実行可能である。

【0022】前記マイクロプロセッサにおいて、VLIW命令の実行を禁止したり、実行を制御するレジスタをプロセッサ内部に有し、VLIW命令実行禁止時にはVLIW命令と同等の動作を行うRISC命令列に分岐することで、VLIWコアの動作を停止させ低消費電力化を図る機能を有する。

【0023】前記VLIWテーブルを不揮発性メモリに内蔵する。前記不揮発性メモリは、マスクROM、フラッシュメモリ、強誘電体メモリのうちのいずれか1つであるのが望ましい。前記VLIWテーブルを揮発性メモリに内蔵する。前記VLIWテーブルを揮発性メモリは、SRAM或いはDRAMのうちのいずれかであるのが望ましい。

【0024】前記VLIWテーブルをRISCコアのプロセッサで書き換え可能なメモリに内蔵するのが望ましい。

【0025】

【発明の実施の形態】図1には本発明の実施形態に係るマイクロプロセッサ100のブロック図が示される。マイクロプロセッサ100は、RISC命令の処理を実行するRISCコア部106、VLIW命令の処理を実行するVLIWコア部140及びデータキャッシュ129から構成される。

【0026】RISCコア部106は、命令フェッチユニット101、命令キャッシュ107、RISCデコードユニット114、浮動小数点ユニット118、整数演算ユニット119、ロード/ストアユニット127から構成される。

【0027】VLIWコア部140は、VLIW起動判定ユニット104、VLIWテーブル113、VLIWデコードユニット115、N個の演算ユニット120、VLIWロード/ストアユニット131から構成される。

【0028】命令フェッチユニット101が命令アドレスバス105に実行する命令のアドレスを出力する。RISC命令実行時には命令アドレスバス105のアドレスに対応する命令が命令キャッシュ107から命令バス108に読み出され、RISCデコードユニット114においてRISC命令のデコードを行う。なお、命令キャッシュ107に該当する命令が存在しない（キャッシュミス）ときは、アドレスバス134とデータバス135を用いてバスコントローラ（図示せず）を介してマイクロプロセッサ外部のメモリから命令が読み出される。

RISCデコードユニット114でのデコードの結果はRISC演算ユニット制御信号117を通して浮動小数点ユニット118や整数演算ユニット119などの制御が行われる。図1ではVLIWコアの演算ユニット140とRISCコアの演算ユニット106をお互いに共有している状態を示しているため、演算ユニット120

へのアクセスが発生することもある。

【0029】さらに命令がメモリアクセス命令であった場合はロード/ストアユニット127へアクセスが発生し、アドレスバス126からデータキャッシュ129にアクセスが起こり、データバス128、RISC用データバス121を通してデータが整数演算ユニット119と浮動小数点ユニット118のレジスタに格納される。データキャッシュ129に該当するデータがない（キャッシュミス）とき、バスコントローラを介して、アドレスバス139とデータバス138を用いて、マイクロプロセッサ外部のメモリをアクセスする。

【0030】RISC命令の実行中にVLW命令の実行を行うための分岐命令（RISC命令の一部）が発生した場合、RISCデコードユニット114からVLW分岐命令デコード信号142を通して命令フェッチユニット101へ分岐発生を知らせる。信号分岐先アドレスは分岐先アドレス用バス136を通して伝えられる。レジスタファイル内のレジスタに分岐先アドレスが格納されている場合は分岐先アドレス用バス141を通して整数演算ユニット119などからRISCデコードユニット114へアドレスが転送される。VLW命令はRISCコアのメモリ空間の特定のアドレス空間に割り付けられており、そのアドレス空間をアクセスすることでVLWテーブル113に格納されているVLW命令をアクセスすることが可能となる。これによって、VLWテーブル113からのVLW命令の読み出し及び書き換えを行うことができる。

【0031】VLW分岐命令デコード信号142を通して分岐の発生を受けた命令フェッチユニット101はVLW命令分岐信号103を通してVLW起動判定ユニット104へVLW命令への分岐が発生したことを知らせる。VLW起動判定ユニット104はVLW命令を起動する条件を判定し、RISCコア制御信号109を通しRISCデコードユニット114へ命令アドレスバス105のアドレスで読み出した命令の実行を中止させる。RISCデコードユニット114内又はVLWデコードユニット115内のレジスタで指定された場合は、消費電力低減のためRISCコア106全体或いは一部のクロックを停止することができる。なお、VLW命令を実行しない場合も、同様にVLWコア全体或いは一部のクロックを停止することができる。

【0032】VLWテーブル113から命令アドレス命令アドレスバス105のアドレスを用いVLW命令バス112へVLW命令が読み出される。VLWデコードユニット115はVLW命令バス112のVLW命令をVLW起動判定ユニット104からVLW命令デコード起動信号111を通して送られてくる起動条件をもとにデコードし、VLW演算ユニット制御信号116を通してVLW演算ユニット120またはRISCコア部106の整数演算ユニット119や浮動

小数点ユニット118のそれぞれの制御を並列に行う。VLW演算ユニット120とは整数演算ユニット、浮動小数点演算ユニット、DSP（Digital Signal Processor:例えば積和演算）等の演算ユニットである。図1には各演算ユニットが個別にレジスタファイルを持つローカルレジスタファイル型のプロセッサの例が示される。各演算ユニット間のデータのやりとりは演算器間データバス123を通して行われる。VLW命令にメモリアクセス命令が含まれていた場合、VLWロード/ストアユニット131がデータキャッシュ129に対しアドレスバス125とデータバス130、VLW用データバス124を用いてアクセスが発生する。

【0033】VLW命令が連続して続く場合は命令アドレスバス105を通してVLW起動判定ユニット104内に保持されているアドレス情報を使い命令アドレスバス132のアドレスとVLWテーブル制御信号133を通してVLWテーブル113からVLW命令が読み出される。読み出されたVLW命令が最後の命令であることを示す完了コードを含んでいる場合VLWデコードユニット115がVLW命令実行状態信号110を通してVLW起動判定ユニット104にVLW命令列の完了を知らせる。VLW命令実行状態信号110にはVLW命令の完了の他に割り込み、例外等のVLW命令実行に影響を与える情報も出力される。VLW起動判定ユニット104はVLW命令列の完了を受けてVLW命令完了信号102とRISCコア制御信号109を通しVLW命令の完了を命令フェッチユニット101とRISCデコードユニット114へ伝え、RISC命令に復帰する。

【0034】すなわち、RISC命令の実行にはRISCコア106を用い、VLW命令の実行にはVLWコア140を用いる。VLWコア140の実行では新規にRISC命令に追加したVLW命令への分岐命令もしくは特定のアドレスへの分岐をトリガとし、VLW命令起動判定ユニット104がマイクロプロセッサ内部に保持するVLWテーブル113から順次VLW命令をVLWデコード115へ読み出す。VLW命令実行後、処理をRISC命令列に復帰する。つまりVLW命令の実行をサブルーチン化しVLWテーブル113に格納することにより高速動作が必要な処理をVLW命令で実行することが可能となる。サブルーチン化することによりVLW回路を有しないマイクロプロセッサであっても、分岐先にVLW命令と同等の動作をするRISC命令列を配置することで互換性の維持が可能である。

【0035】また低消費電力動作が必要であり、かつVLW命令を使うような高速な動作が必要でない場合には内蔵制御レジスタの状態により、VLWコア140の動作を停止し、RISCコア106のみで動作を行うことも可能である。逆にRISCコア106の動作が必

要でない場合にはRISCコア106の動作を停止し、VLIWコア140のみで動作することも可能である。

【0036】さらに、RISC命令の実行中にVLIWコア140の演算器(レジスタファイルを含む)を使用でき、VLIW命令実行中にRISCコア106の演算器(レジスタファイル)を使用できるので、資源の共用が可能になり半導体集積回路のチップ面積低減が可能になる。

【0037】VLIW起動判定ユニット104は、VLIW命令実行中に実行中のVLIW命令がVLIW命令実行完了後に復帰する先のRISC命令と依存性がないかを制御信号137を通して得られるデコード結果をもとに判定し、依存性がない場合にVLIW命令完了信号102とRISCコア制御信号109を通してRISC命令とVLIW命令の並列実行の制御の選択が可能である。

【0038】VLIWテーブル113にはマスクROM、EPROM(電気的書き込み紫外線消去の不揮発性メモリ)、EEPROM(電気的消去/書き込み可能な不揮発性メモリ)、フラッシュメモリ(フラッシュEEPROM)、FRAM(強誘電体メモリ)等の不揮発性メモリと、DRAM(ダイナミックRAM)、SRAM(スタティックRAM)等の揮発性メモリとが使用される。書き換え可能なメモリ、例えばEEPROM、フラッシュメモリ、FRAM、DRAM、SRAM等を用いた場合、VLIWテーブルデータバス122を通してシステムに応じたVLIWテーブルを設定することが可能である。これによって、VLIW動作中に動的にVLIWテーブルをシステムに最適な構成にすることができる。またVLIW起動判定ユニット104内にあるレジスタへのアクセスにもこのVLIWテーブルデータバス122が用いられる。SRAMを用いると他のメモリを使用する場合に比べて書き込み及び読み出しが高速にできる。DRAMはメモリ素子が1個のトランジスタで構成され、SRAMはメモリ素子が4個又は6個のトランジスタで構成されるため、DRAMを用いることでVLIWテーブルをSRAMを用いるより小さくできる。図2には図1に係るマイクロプロセッサの変形例が示される。図2ではRISCコア106とVLIWコア140のレジスタを共通に持つグローバルレジスタ型のマイクロプロセッサが示される。図1のマイクロプロセッサ100と同一の部分には同一の記号が用いられている。

【0039】図5には、VLIW分岐命令によりVLIW命令がRISC命令と並列に実行する場合のタイミングチャートとVLIW命令のコード割付の例が示されている。

【0040】VLIW分岐命令はRISC命令の1つで、RISC命令1、2、3と同様にRISCコア106でパイプライン処理で実行される。記号501、50

4、506、507はそのパイプラインステージを示している。Iは命令フェッチ、Dは命令デコード、Eは実行、Mはメモリアクセス、Wは書き込みのステージを表す。また、VLIW命令1、2、3は、VLIWコア140でパイプライン処理で実行される。記号502、503、505はそのパイプラインステージを示している。パイプラインのVI、VD、VE、VM、VWはそれぞれVLIW用の命令フェッチ、命令デコード、実行、メモリアクセス、書き込みのステージを表す。

【0041】VLIW分岐命令からVLIW命令1へ3サイクル後に分岐するとしたらVLIW命令1のパイプライン502がサイクル4から開始する。VLIW命令2以降のVLIW命令がVLIW命令実行後の復帰先のRISC命令と依存性がない場合、VLIW命令2の実行503とRISC命令1の実行504を並列に行うことが可能である。VLIW命令3の実行505とRISC命令2の実行506も並列に動作でき、VLIW命令が完了すると、VLIW命令用のパイプラインは消え、RISC命令3の実行507が行われる。

【0042】VIステージでVLIWテーブルから読み出されるVLIW命令のコード割付の例を図5の下に示す。VLIW命令には複数の演算器を制御するためのコードが1命令に埋めこまれている。このVLIW命令の命令長は128ビット或いは256ビットの固定長である。記号508はVLIW演算器1を制御するためのフィールド、509はVLIW演算器2を制御するためのフィールド、510は整数演算器を制御するためのフィールド、511は浮動小数点演算器を制御するためのフィールド、512はVLIW命令の完了やその他の情報を格納するためのフィールドなどから構成されている。508、509、510、511、512の各々のフィールドは32ビット長で、513のオペコード、514のディスプレイメント、515のソースレジスタ1、516のソースレジスタ2、517のデスティネーションレジスタなどの割付が行われている。この命令コードをVLIWデコーダがデコードし各演算器を制御する。

【0043】本発明の実施形態では、命令セットの異なるプロセッサ間の動作を切り替える方法として、プロセッサを切り替えるための分岐命令若しくは特定アドレス空間への分岐をトリガとして用いている。第1のプロセッサと第2のプロセッサの間で動作を切り替える場合、第1のプロセッサ若しくは第2のプロセッサのどちらかのプロセッサが全体の実行を制御するマスタプロセッサとなり、もう一つがスレーブプロセッサとなる。マスタプロセッサは常に片方のプロセッサに固定であるわけではなく、動的にどちらのプロセッサでもマスタプロセッサになることは可能である。

【0044】マスタプロセッサからスレーブプロセッサへ動作が移った後、スレーブプロセッサの動作が復帰後

のマスタープロセッサの動作に影響を与えない場合、スレーブプロセッサもしくはマスタプロセッサが依存性を判断することで並列に2つのプロセッサを動作させることが可能である。

【0045】図6には、本発明を利用したシステムの全体図が示される。601は1枚或いは複数枚のプリント基板上に構成されたシステムで、このシステムにディスプレイ611、CD-ROM612、ビデオカメラ615、LCD618、キーボードやプリンタ等の周辺デバイス620等が接続される。システム601はプロセッサ602、プログラム等を格納するROM608、データやプログラムを格納するSRAM609、DRAM614及びSDRAM614、表示用画像データ等を格納する3次元グラフィック用フレームバッファ610、CD-ROMを制御するCD-ROMデコーダ613、ビデオカメラを制御するビデオコントローラ616、液晶表示装置(LCD)を制御するLCDコントローラ619、キーボードやプリンタ等の周辺デバイス620を制御する周辺デバイスコントローラ621等がシステムバス607に接続されて構成されている。

【0046】プロセッサ602は、第1の命令セットを持つプロセッサ1(603)と、第2の命令セットを持つプロセッサ2(605)から構成されており、制御信号606によりプロセッサ1がプロセッサ2を、制御信号604によりプロセッサ2がプロセッサ1を制御する。プロセッサ602は3次元グラフィックス、MPEGのエンコード/デコードを高速に処理することができるので、従来のような専用のハードウェア回路が必要なく、プリント基板上に実装する半導体チップの個数を削減でき、プリント基板の枚数の削減或いはプリント基板自身の大きさを小さくできる。

【0047】図7には、図6のプロセッサ602を一つの半導体集積回路装置で構成した(ワンチップに内蔵した)場合のブロック図が示されている。プロセッサ701は、半導体集積回路製造技術を用いて単結晶シリコンのような1個の半導体基板上に形成される。この半導体基板はレジンモールド技術を用いていわゆるプラスチックパッケージに封止される。このワンチップには、ROM608やRAM609などのメモリが含まれることもある。

【0048】ワンチップのプロセッサ701は、第1の命令セットを持つプロセッサ1(708)と、第2の命令セットを持つプロセッサ2(711)、データキャッシュ712、周辺装置、クロックパルス発生回路(CPG)714、入出力回路(I/O)715等から構成される。

【0049】プロセッサ1(708)とプロセッサ2(711)の組み合わせには16ビット固定長命令のRISCプロセッサと64ビット固定長命令のVLIWプロセッサの組み合わせや、32ビット固定長命令のRISC

プロセッサと128ビット固定長命令のVLIWプロセッサの組み合わせや、64ビット固定長命令のRISCプロセッサと256ビット固定長命令のVLIWプロセッサ等がある。この組み合わせではプロセッサ1とプロセッサ2を逆にすることも可能である。図7ではプロセッサ1(708)がRISCプロセッサ(図1のRISCコア106)、プロセッサ2(711)がVLIWプロセッサ(図1のVLIWコア140)である場合が示される。

【0050】プロセッサ1(708)は命令フェッチユニット101、命令キャッシュ107、RISCデコードユニット114、1つまたは複数の演算ユニット704、RISCロード/ストアユニット127から構成され、内部バス702により接続されている。命令フェッチユニット101が命令キャッシュ107から命令を読み出し、その読み出した命令をRISCデコードユニット114で解析し、演算ユニット704が動作し、メモリアクセスが必要な場合はロード/ストアユニット127が起動する。演算ユニット704には乗算ユニット、整数演算ユニット、浮動小数点ユニット、除算ユニット、DSPなどの演算ユニットが置かれる。また同一の演算ユニットを複数個持つ場合もある。命令キャッシュへの命令の読み込みはバス709を用いて、バスコントロールユニット712から行われる。RISCロード/ストアユニット127はバス705を用いてデータキャッシュ129にアクセスを行う。データキャッシュ129へのデータの取り込みはバスコントロールユニット712からバス707を用いて行われる。

【0051】プロセッサ2(711)はVLIW起動判定ユニット104、VLIWテーブル113、VLIWデコードユニット115、1つまたは複数の演算ユニット704、VLIWロードストアユニット131から構成され、内部バス702により接続されている。VLIW起動判定ユニット104からVLIW命令の起動がかけると、VLIWテーブル113よりVLIW命令が読み出され、VLIWデコードユニット115で命令を解析し演算ユニット704が動作し、メモリアクセスが必要な場合はロード/ストアユニット131が起動する。VLIWテーブル113にはマスクROMやフラッシュメモリなどの不揮発性メモリ又はSRAMやDRAMなどの揮発性メモリが使用される。VLIWテーブルはチップ外に存在することもあり、その場合にはバスコントロールユニットを通して外部からVLIW命令の読み出しが行なわれる。演算ユニット704には乗算ユニット、整数演算ユニット、浮動小数点ユニット、除算ユニット、DDSPなどの演算ユニットが置かれる。また同一の演算ユニットを複数個持つ場合もある。VLIWロード/ストアユニット131はバス706を用いてデータキャッシュ129にアクセスを行う。

【0052】バスコントロールユニット712はバス7

13を用いてI/O715にアクセスを行う。I/O715はアドレス信号717、データ信号716を用いてチップの外とのデータのやりとりを行う。バス713には周辺モジュール710等が接続されることもある。また内部のクロックはCPG714によって発生される。

【0053】プロセッサ1(708)とプロセッサ2(711)とのあいだのインターフェイスは制御信号線(又はバス)703を用いて行われる。

【0054】図7のようにプロセッサ1とプロセッサ2をワンチップ化することにより、データキャッシュを共有できるためデータの一致制御が行いやすくなる。またプロセッサ間の信号がチップの外に出ないためセキュリティが強化される。さらにプロセッサ間の通信の高速化と情報量が削減できるため、低コスト化と高速化が図れる。

【0055】図8には本発明に係る他の実施形態であるマイクロコンピュータ800のブロック図が示される。マイクロコンピュータ800は半導体製造プロセス技術を用いてシリコンのような1つの半導体基板上に形成される。この半導体基板はレジンモールド技術を用いていわゆるプラスチックパッケージに封止される。

【0056】マイクロコンピュータ800は、第1のプロセッサ801、第2のプロセッサ802、第1のメモリ803、第2のメモリ804、プロセッサ切り替え回路805、アドレスバス806、データバス807、及びバス制御ユニット808を有する。

【0057】第1のプロセッサ801はRISC型プロセッサで、16ビット固定長命令を実行するプロセッサである。第2のプロセッサ802はVLIW型のプロセッサで、64ビット長命令を実行するプロセッサである。第1のプロセッサ801と第2のプロセッサ802との命令体系(命令セット、命令フォーマット或いはプログラミングモデル等)は異なる。第1のメモリ803には第1のプロセッサ801が実行する命令(プログラム)が格納される。第2のメモリ804には第2のプロセッサ802が実行される命令(プログラム)が格納される。第1のメモリ803及び第2のメモリ804は不揮発性のメモリ又は揮発性のメモリで構成される。第1のメモリ803と第2のメモリ804を書き替え可能なメモリで構成することにより、半導体供給者だけでなく、半導体の使用者がプログラムすることが可能になる。すなわち、第1の命令セットを用いたプログラムと第2の命令セットを用いたプログラムとの両方が使用者によってすることができる。不揮発性のメモリ、揮発性のメモリ及び書き替え可能なメモリには、図1のマイクロプロセッサのVLIWテーブル113で使用されるものと同様である。

【0058】図9のメモリマップに示すように第1のプロセッサのプログラムと第2のプロセッサのプログラム

は、第1のプロセッサ801或いは第2のプロセッサ802のアドレス空間上にある。第1のメモリ803には第1のプロセッサのプログラムの一部或いは全てが格納される。また、第2のメモリ804には第2のプロセッサのプログラムの一部或いは全てが格納される。

【0059】プロセッサ切り替え回路805は、第1のプロセッサ801或いは第2のプロセッサ802のアドレスが入力され、第1のメモリ803と第2のメモリ804のどちらをアクセスするのかを判定する。第1のプロセッサ801が第1のメモリ803をアクセスする場合は、アドレス信号と制御信号812をプロセッサ切り替え回路805に入力する。プロセッサ切り替え回路805は第1のメモリ803のアクセスに必要なアドレス信号や制御信号808を出力する。また、第2のプロセッサ802に先読みされた命令の動作の中止や低消費電力化のためのクロック停止のための制御信号810を出力する。一方、第2のプロセッサ802が第2のメモリ804をアクセスする場合は、アドレス信号と制御信号813をプロセッサ切り替え回路805に入力する。プロセッサ切り替え回路805は第2のメモリのアクセスに必要なアドレス信号や制御信号を信号線809に出力する。また、第1のプロセッサ801に先読みされた命令の動作の中止や低消費電力化のためのクロック停止のための制御信号811を出力する。プロセッサ切り替え回路805によって、第1のプロセッサと第2のプロセッサが排他的に動作する。

【0060】第1のプロセッサ801は第1のプロセッサのプログラムを実行し、そのプログラムの中の分岐命令等で第2のプロセッサのプログラム領域を示すアドレスを出力することによって、第2のプロセッサで第2のプロセッサのプログラムを実行できる。また、第2のプロセッサは第2のプロセッサのプログラムの中の分岐命令等で第1のプロセッサのプログラム領域を示すアドレスを出力することによって、第1のプロセッサのプログラムに戻るることができる。

【0061】なお、プロセッサ切り替え回路805内には制御レジスタがあり、このレジスタを書き替えることにより図9に示したメモリマップは変更することができる。前記制御レジスタで、第1のプロセッサのプログラム領域、第2のプロセッサのプログラム領域、データ領域のそれぞれの始まりアドレスと終わりのアドレス等を指定できる。

【0062】これによって、命令体系の異なるプログラムが1つのマイクロコンピュータで実行でき、第1のプロセッサのプログラムが従来から蓄積されたソフトウェアであっても、新しいアーキテクチャのプログラムが第2のプロセッサで実行できるので、ソフトウェアの互換性が維持できる。

【0063】また、動作しないプロセッサのクロックを停止させることで、低消費電力化を実現できる。

【0064】第1のプロセッサ801と第2のプロセッサ802の命令長は比較的短いため、命令デコーダ等の制御部や演算部を小さくできるので、第1のメモリ803又は第2のメモリ804の容量を大きくすることができる。なお、プロセッサの面積が大きいても良い場合は、命令長は長くしてもよい。

【0065】本発明の実施の形態により、命令セットの異なるプロセッサ間の動作を特定の命令で切り替えることが可能となるため、互換性を維持しながら異なる命令セットを有するプロセッサへの拡張が可能となる。

【0066】またプロセッサ間で演算ユニットなどの資源を共有化できるため、低コスト化につながる。

【0067】異なる命令セットを持つプロセッサをワンチップに内蔵した場合には、プロセッサ間通信の高速化と、内部情報がチップ外部に出力されないことによるセキュリティ強化が図れる。

【0068】動作中に高速な処理が必要でなくなったとき1つのプロセッサの動作を完全に止め、別のプロセッサで止めたプロセッサと同機能の処理をさせることにより低消費電力化が図れる。

【0069】これによって、(1)ソフトウェアは従来と完全互換性を保ったまま1GIPS (Giga Instructions Per Second)、最大10GOPS (Giga Operations Per Second) の高性能を実現できる。(2)既存のソフトウェアのままでミドルウェア部分の性能を1桁向上できる。(3)新しいアプリケーションもこれまでの命令セットのままで作成できる。(4)全てのアプリケーションをソフトウェアで実現できる。

【0070】以上本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることはいうまでもない。例えば、図1のマイクロプロセッサでは、RISCコアの演算器とVLIWコアの演算器とは双方で共用できたが、図2のマイクロプロセッサのように独立させてもよい。図2ではロード/ストアユニット204RISCコアとVLIWコアとの共有にした場合をしたが、ロード/ストアユニットは図1のように2つ持っても構わない。第1のプロセッサをRISC型プロセッサ、第2のプロセッサをVLIW型プロセッサで構成したが、命令セットの異なる2つのプロセッサで構成することができる。すなわち、RISC型とRISC型、RISC型とCISC型、CISC型とCISC型、CISC型とVLIW型、VLIW型とVLIW型等のプロセッサを用いても良い。また、RISCプロセッサとVLIWプロセッサとの命令長を16ビットと64ビット、32ビットと128ビット、64ビットと256ビットとの例を示す。

【0071】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記

の通りである。

【0072】すなわち、従来のソフトウェアを実行する第1のプロセッサと、新しいアーキテクチャのソフトウェアを実行する第2のプロセッサとを有し、第1のプロセッサが第2のプロセッサに起動をかけることができるので、従来のソフトウェアと互換性を維持しながら、新しいアーキテクチャ(命令体系)のソフトウェアを実行する高性能なプロセッサを実現することができる。

【0073】2つのプロセッサ内部の資源たとえば演算ユニットなどを共有化することにより、ハードウェアの増加を抑えて高速性能を実現することができる。

【0074】2つのプロセッサを並列に動作させる場合と、間欠動作させる2つの制御方式が可能であるため、高速化と低消費電力化の双方を実現することができる。

【図面の簡単な説明】

【図1】本発明の一つの実施形態に係るマイクロプロセッサのブロック図。

【図2】図1に係るマイクロプロセッサの変形例のブロック図。

【図3】図1及び図2のマイクロプロセッサのVLIW起動論理に関する詳細図。

【図4】図1及び図2のマイクロプロセッサにTLBを用い場合のVLIW起動論理に関する詳細図。

【図5】VLIW命令実行時のパイプラインの構成と命令コードの例。

【図6】本発明に係るマイクロプロセッサを利用したシステムの全体図。

【図7】ワンチップのプロセッサのブロック図。

【図8】本発明に係る他の実施形態であるマイクロコンピュータのブロック図。

【図9】図8のマイクロコンピュータのメモリマップ。

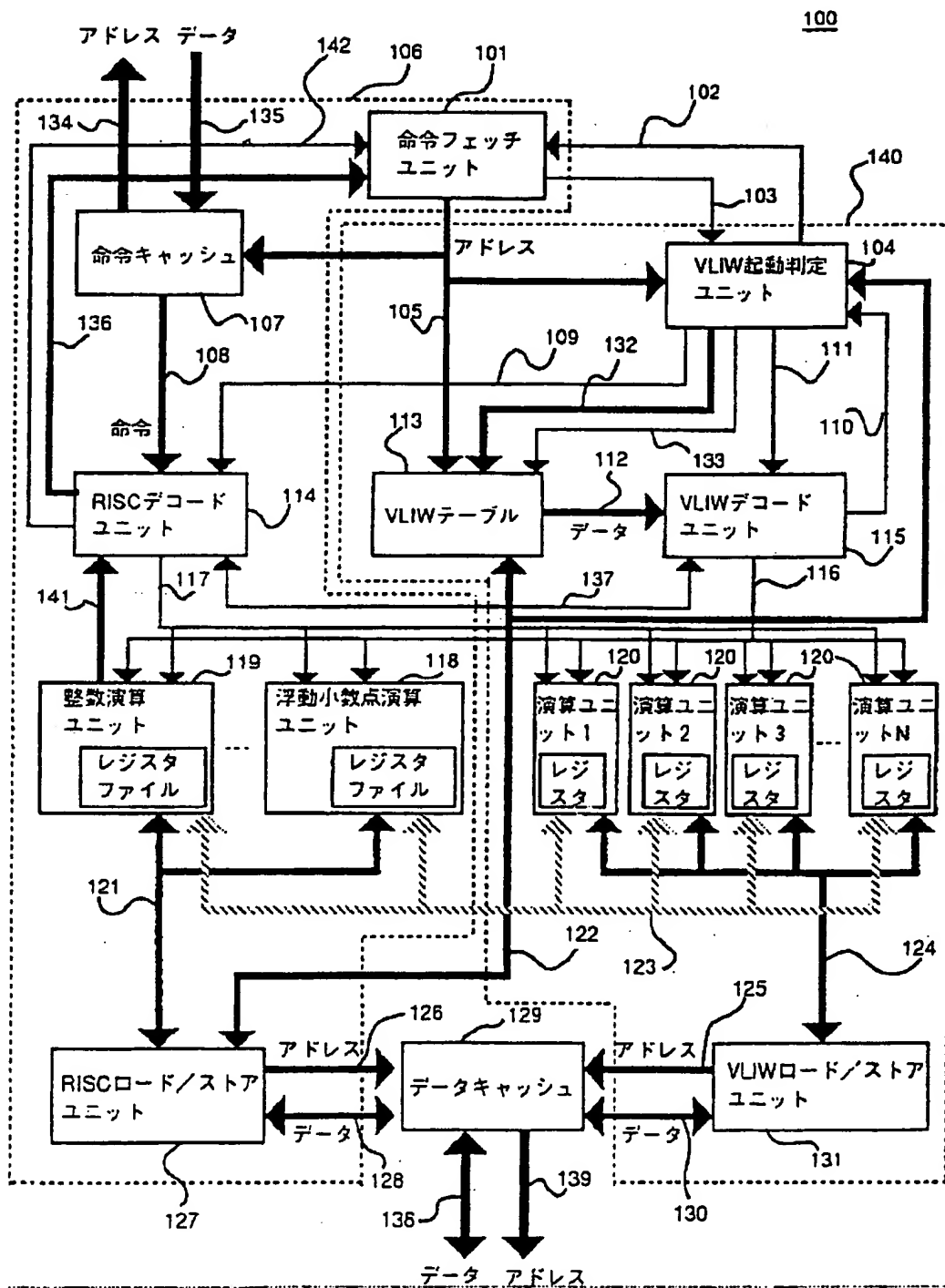
【符号の説明】

- 100 マイクロプロセッサ
- 101 命令フェッチユニット
- 102 VLIW命令完了信号
- 103 VLIW命令分岐信号
- 104 VLIW起動判定ユニット
- 105、132 命令アドレスバス
- 106 RISCコア
- 107 命令キャッシュ
- 108 命令バス
- 109 RISCコア制御信号
- 110 VLIW命令実行状態信号
- 111 VLIW命令デコード起動信号
- 112 VLIW命令バス
- 113 VLIWテーブル
- 114 RISC命令デコードユニット
- 115 VLIW命令デコードユニット
- 116 VLIW演算ユニット制御信号
- 117 RISC演算ユニット制御信号

118	浮動小数点演算ユニット	504	RISC命令1
119	整数演算ユニット	505	VLIW命令3
120	VLIW演算ユニット	506	RISC命令2
121	RISC用データベース	507	RISC命令3
122	VLIWテーブルデータベース	508	VLIW演算器1制御フィールド
123	演算器間データベース	509	VLIW演算器2制御フィールド
124	VLIW用データベース	510	整数演算器制御フィールド
125、126	アドレスバス	511	浮動小数点演算器制御フィールド
127	RISCロード/ストアユニット	512	VLIW命令完了コード
128、130	データベース	513	オペコード
129	データキャッシュ	514	ディスプレイメント
131	VLIWロード/ストアユニット	515	ソースレジスタ1
133	VLIWテーブル制御信号	516	ソースレジスタ2
134、139	アドレスバス	517	ディスティネーションレジスタ
135、138	データベース	601	システム
136、141	分岐先アドレス用バス	602	プロセッサ
137	制御信号	603	プロセッサ1
140	VLIWコア	604	プロセッサ2からプロセッサ1の制御信号
142	VLIW分岐命令デコード信号	605	プロセッサ2
200	マイクロプロセッサ	606	プロセッサ1からプロセッサ2の制御信号
201	レジスタファイル	607	システムバス
202	演算ユニットとレジスタファイル間のバス	608	ROM
203	レジスタファイルとロードストアユニット間のバス	609	RAM
204	ロード/ストアユニット	610	3次元グラフィックス用フレームバッファ
205	データベース	611	ディスプレイ
206	アドレスバス	612	CD-ROM
301	加算器	613	CD-ROMデコーダ
302	VLIWコア用プログラムカウンタ	614	DRAM
303	セレクタ	615	ビデオカメラ
304	アドレスラッチ	616	ビデオコントローラ
305	データラッチ	617	SDRAM
306	VLIWテーブル格納メモリ	618	LCD
307	テーブル制御回路	619	LCDコントローラ
308	VLIWコア制御レジスタ	620	キーボードやプリンタなど周辺デバイス
309	デコーダ	621	周辺デバイスコントローラ
310	VLIW起動判定回路	701	プロセッサ
311	調停回路	702	内部バス
312	起動回路	703	プロセッサ1とプロセッサ2の間の制御信号
313	マルチプレクサ	704	演算ユニット(整数演算、浮動小数点演算、乗除算、DSPなど)
314	VLIW演算ユニット制御回路	705	プロセッサ1とデータキャッシュ間のバス
315	VLIW分岐条件判定回路	706	プロセッサ2とデータキャッシュ間のバス
316	RISCコア用プログラムカウンタ	707	データキャッシュとバスコントロールユニット間のバス
317	加算器	708	プロセッサ1(RISCプロセッサ)
401	アドレス変換バッファ(TLB)	709	命令キャッシュとバスコントロールユニット間のバス
402	VLIW起動補助情報信号	710	周辺モジュール
403	命令アドレス(物理アドレス)	711	プロセッサ2(VLIWプロセッサ)
501	VLIW分岐命令のパイプライン	712	バスコントロールユニット
502	VLIW命令1		
503	VLIW命令2		

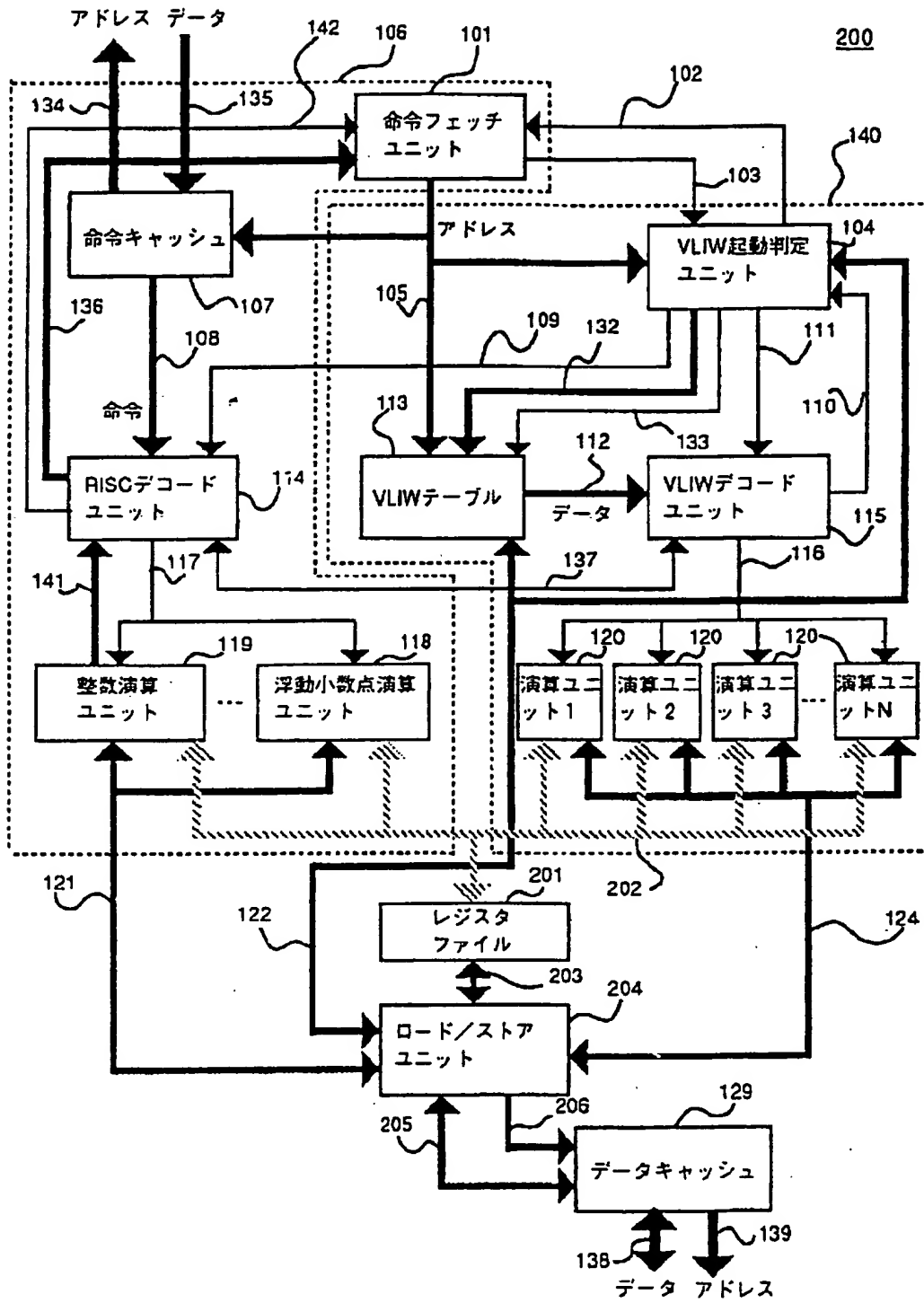
- 802 第2のプロセッサ
803 第1のメモリ
804 第2のメモリ
805 プロセッサ切り替え回路
806 アドレスバス
807 データバス
808 バス制御ユニット。

图 1



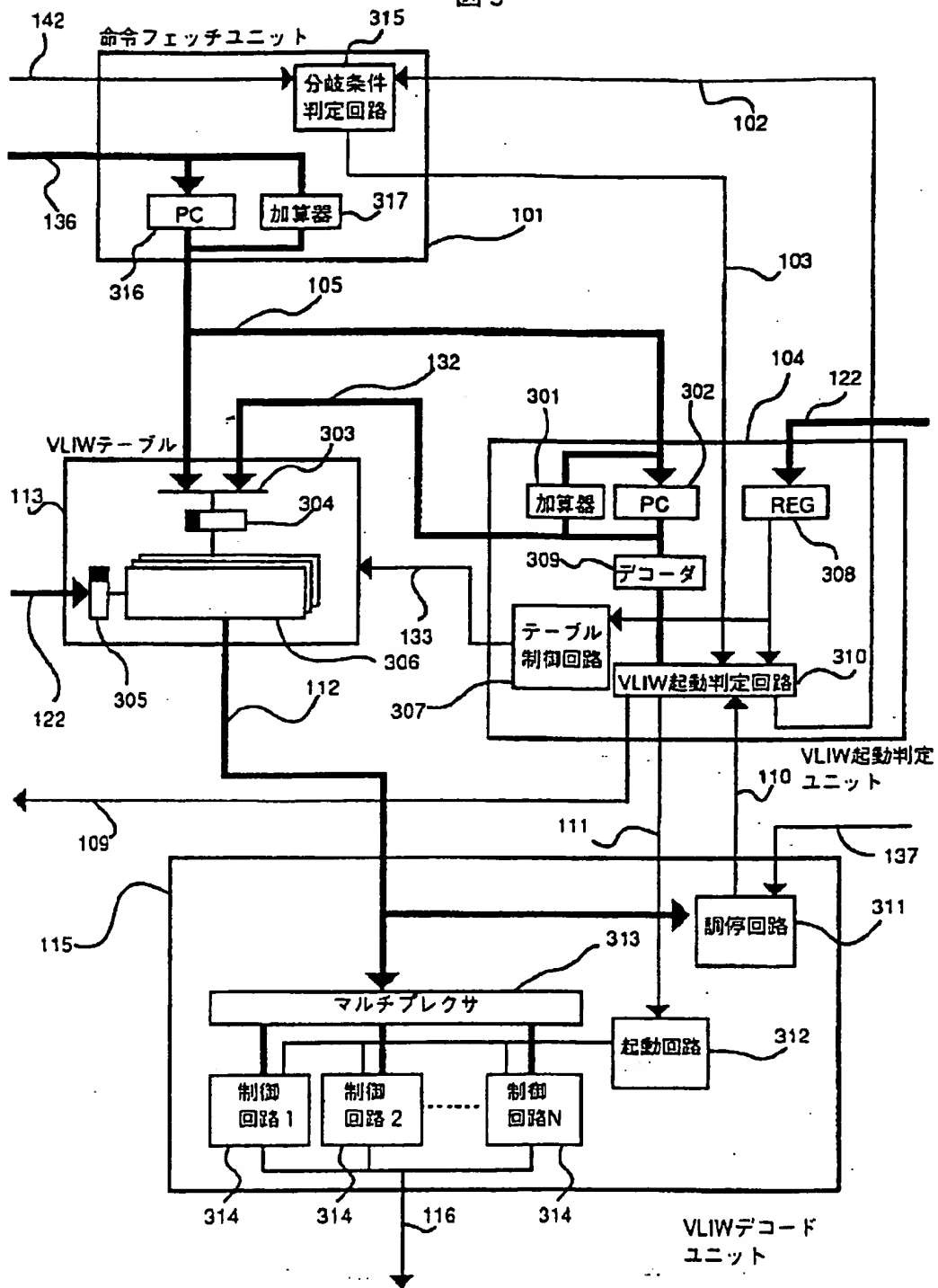
【図2】

図2



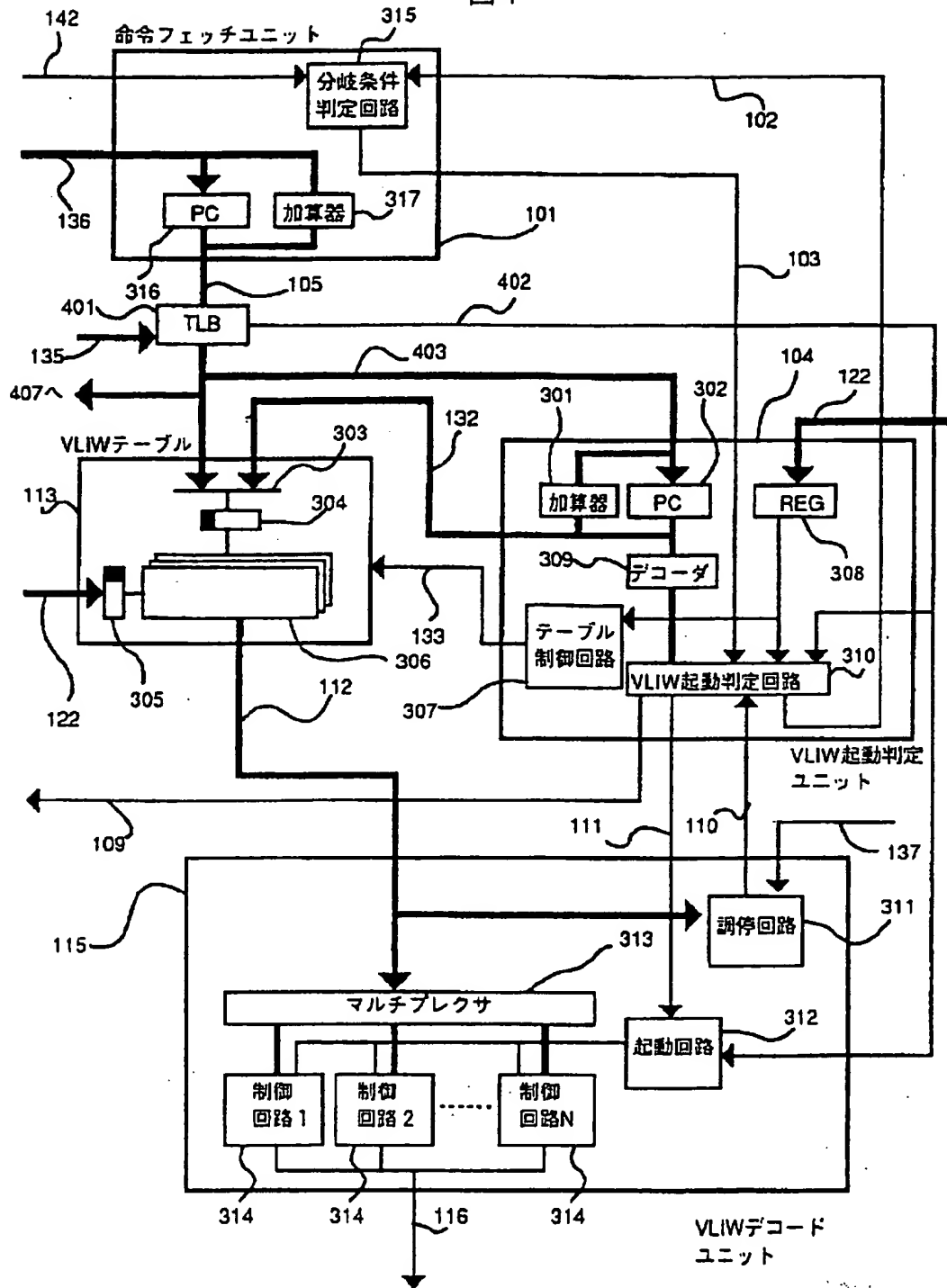
【図3】

図3



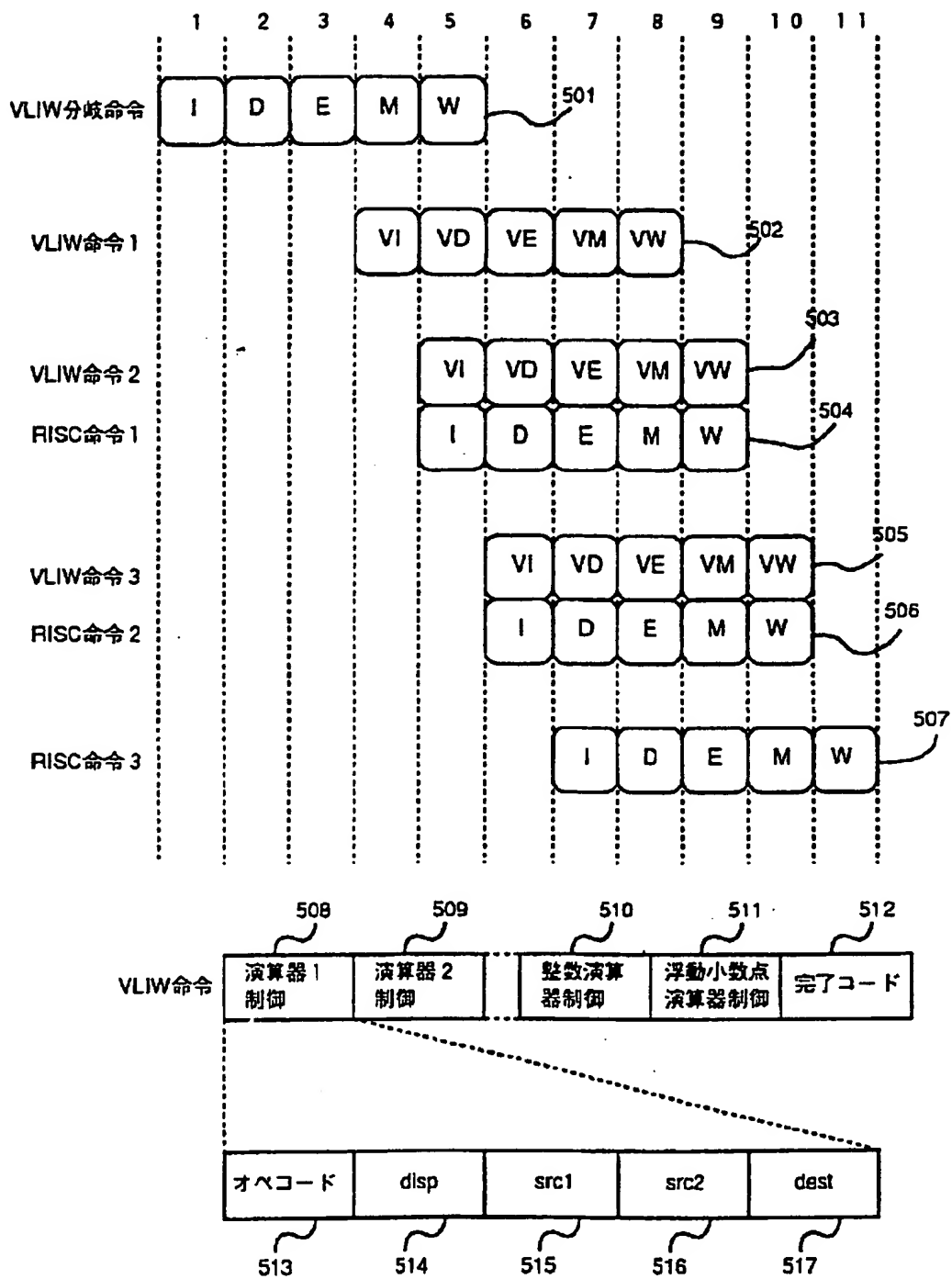
【図4】

図4



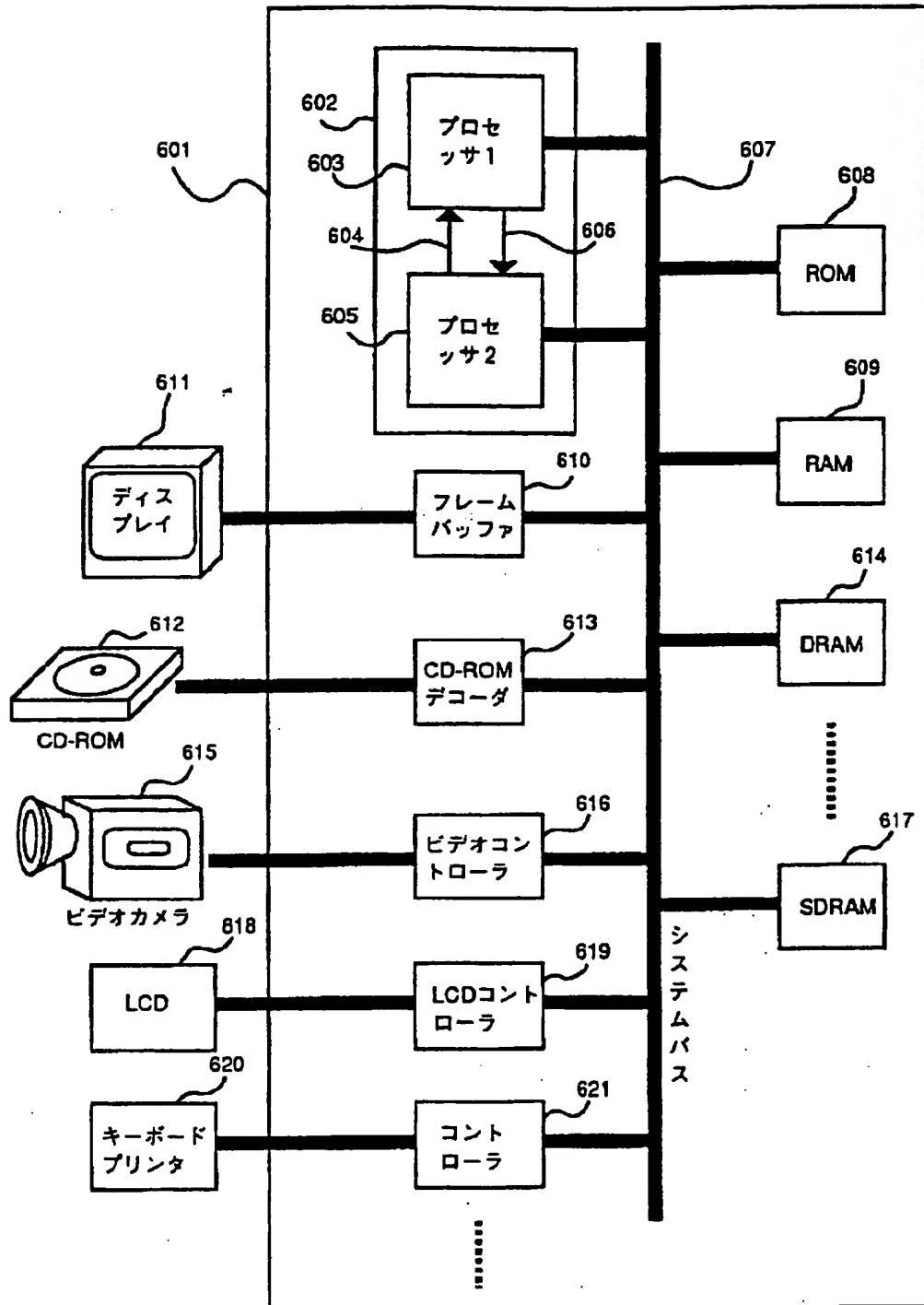
【図5】

図5



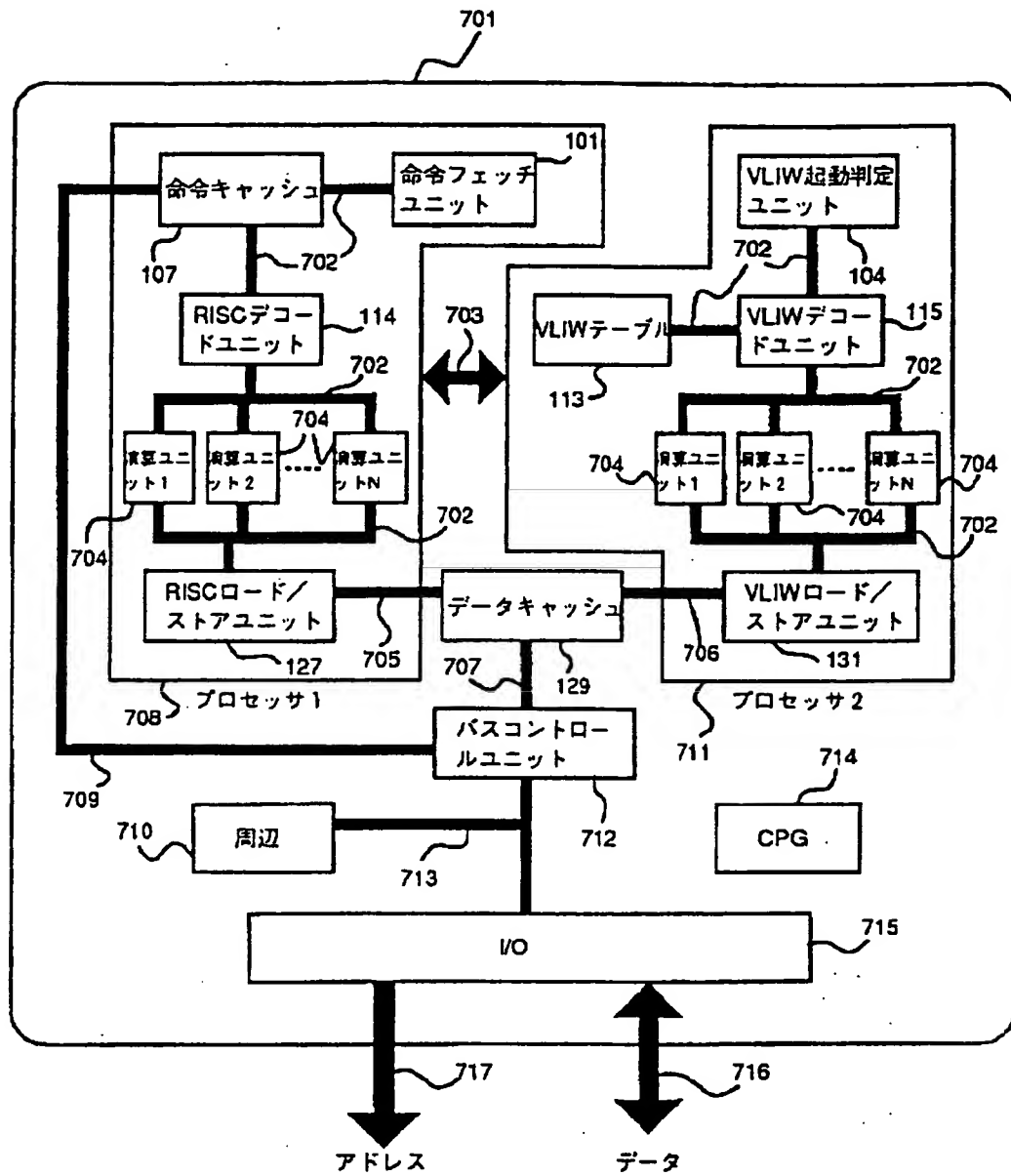
【図6】

図6



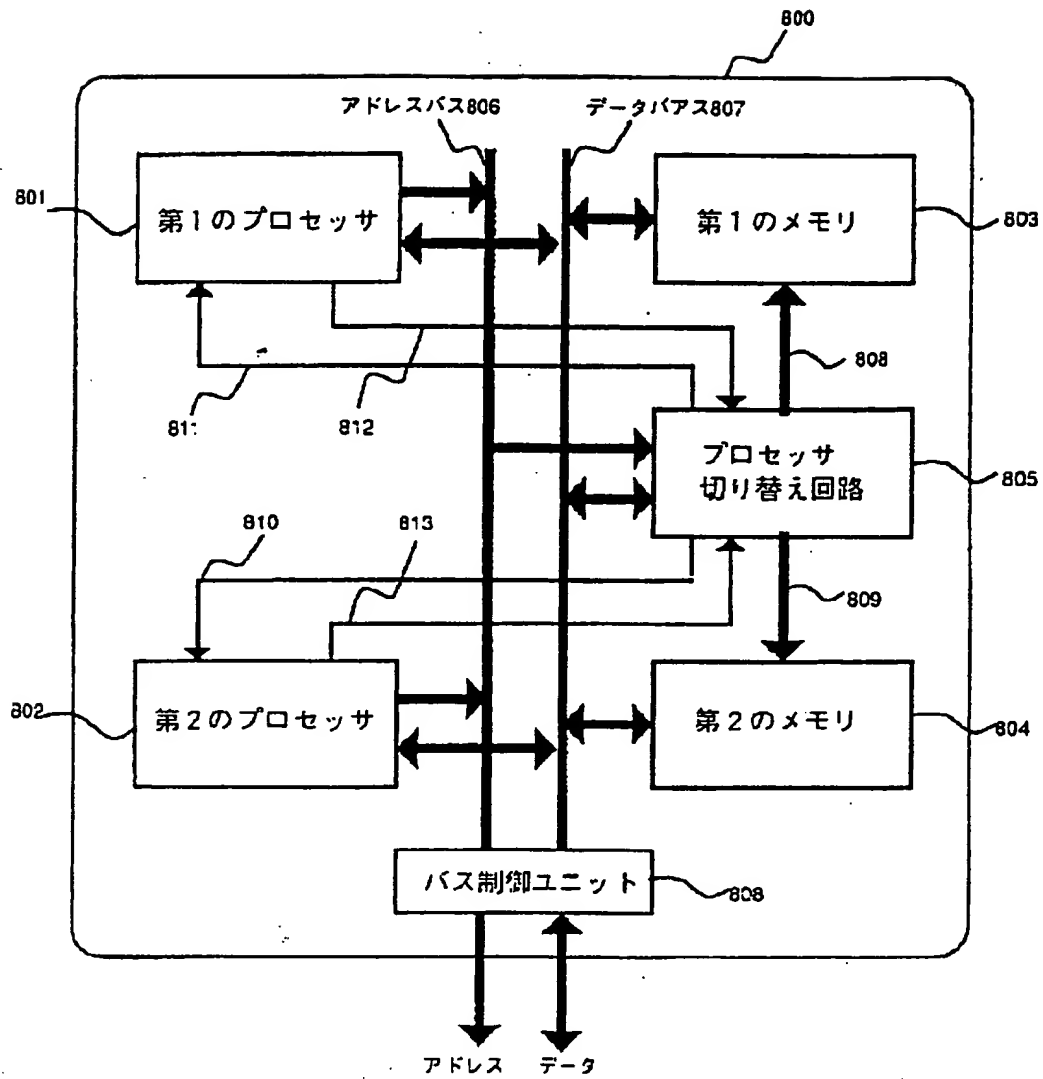
【図7】

図7



【図8】

図8



【図9】

図9

メモリマップ
アドレス空間

